M.Sc. ADILSON OLIVEIRA CRUZ

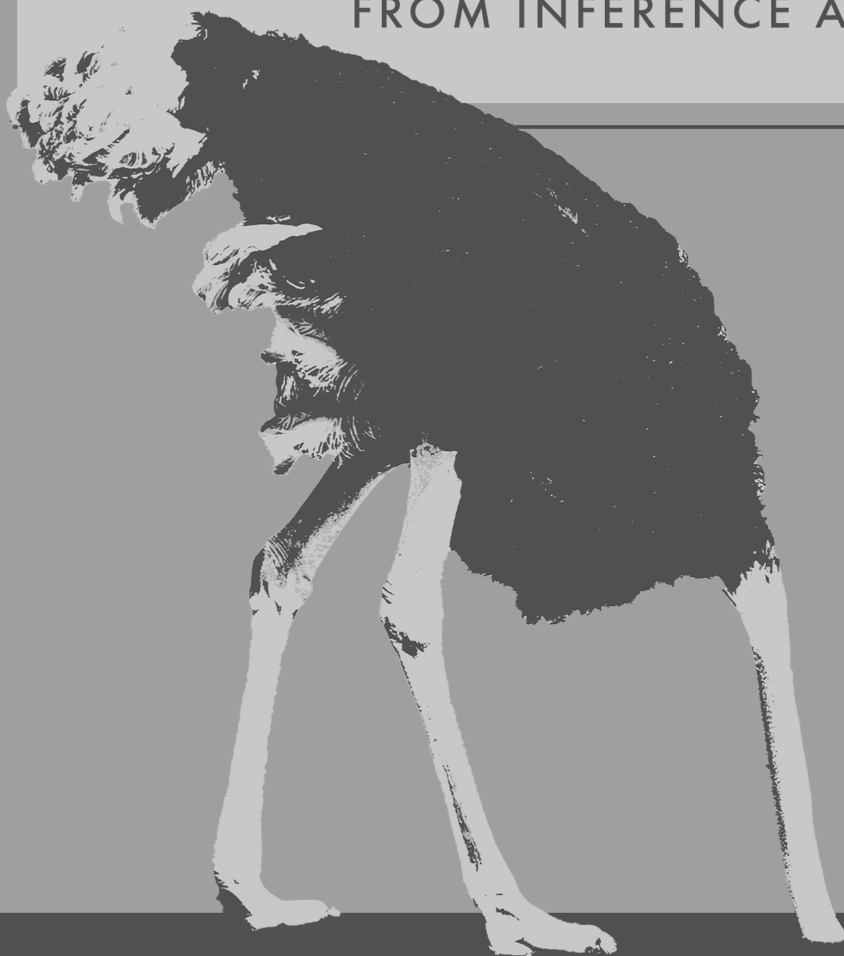
A LBS MIDDLEWARE WITH PRIVACY PROTECTION FROM INFERENCE ATTACKS



Atena Editora
Ano 2023

M.Sc. ADILSON OLIVEIRA CRUZ

A LBS MIDDLEWARE WITH PRIVACY PROTECTION FROM INFERENCE ATTACKS



Atena Editora

Ano 2023

Editora chefe
Profª Drª Antonella Carvalho de Oliveira
Editora executiva
Natalia Oliveira
Assistente editorial
Flávia Roberta Barão
Bibliotecária
Janaina Ramos
Projeto gráfico
Camila Alves de Cremo
Ellen Andressa Kubisty
Luiza Alves Batista
Nataly Evilin Gayde
Thamires Camili Gayde
Imagens da capa
iStock
Edição de arte
Luiza Alves Batista

Todos os manuscritos foram previamente submetidos à avaliação cega pelos pares, membros do Conselho Editorial desta Editora, tendo sido aprovados para a publicação com base em critérios de neutralidade e imparcialidade acadêmica.

A Atena Editora é comprometida em garantir a integridade editorial em todas as etapas do processo de publicação, evitando plágio, dados ou resultados fraudulentos e impedindo que interesses financeiros comprometam os padrões éticos da publicação. Situações suspeitas de má conduta científica serão investigadas sob o mais alto padrão de rigor acadêmico e ético.

CIÊNCIAS EXATAS E DA TERRA E ENGENHARIAS

A LBS middleware with privacy protection from inference attacks

Diagramação: Nataly Evilin Gayde
Correção: Yaiddy Paola Martinez
Indexação: Amanda Kelly da Costa Veiga
Revisão: O autor
Autor: M. Sc. Adilson Oliveira Cruz

Dados Internacionais de Catalogação na Publicação (CIP)

C957 Cruz, Adilson Oliveira
A LBS middleware with privacy protection from inference attacks / Adilson Oliveira Cruz. – Ponta Grossa - PR: Atena, 2023.

Formato: PDF
Requisitos de sistema: Adobe Acrobat Reader
Modo de acesso: World Wide Web
Inclui bibliografia
ISBN 978-65-258-1875-7
DOI: https://doi.org/10.22533/at.ed.757230510

1. Legislation on personal data protection. I. Cruz, Adilson Oliveira. II. Título.

CDD 343.09981

Elaborado por Bibliotecária Janaina Ramos – CRB-8/9166

Atena Editora
Ponta Grossa – Paraná – Brasil
Telefone: +55 (42) 3323-5493
www.atenaeditora.com.br
contato@atenaeditora.com.br

DECLARAÇÃO DO AUTOR

DECLARAÇÃO DA EDITORA

A Atena Editora declara, para os devidos fins de direito, que: 1. A presente publicação constitui apenas transferência temporária dos direitos autorais, direito sobre a publicação, inclusive não constitui responsabilidade solidária na criação dos manuscritos publicados, nos termos previstos na Lei sobre direitos autorais (Lei 9610/98), no art. 184 do Código Penal e no art. 927 do Código Civil; 2. Autoriza e incentiva os autores a assinarem contratos com repositórios institucionais, com fins exclusivos de divulgação da obra, desde que com o devido reconhecimento de autoria e edição e sem qualquer finalidade comercial; 3. Todos os e-book são *open access, desta forma* não os comercializa em seu site, sites parceiros, plataformas de *e-commerce,* ou qualquer outro meio virtual ou físico, portanto, está isenta de repasses de direitos autorais aos autores; 4. Todos os membros do conselho editorial são doutores e vinculados a instituições de ensino superior públicas, conforme recomendação da CAPES para obtenção do Qualis livro; 5. Não cede, comercializa ou autoriza a utilização dos nomes e e-mails dos autores, bem como nenhum outro dado dos mesmos, para qualquer finalidade que não o escopo da divulgação desta obra.

Title/Título: A Lbs Middleware With Privacy Protection From Inference Attacks

Author/Autor: Ms.C. Adilson Oliveira Cruz

Educational institution/Instituição de ensino: Instituto Federal do Espírito Santo (IFES)

Address/Endereço: Av. Rio Branco, 50 - Santa Lucia, Vitória - ES, 29056-264

ACKNOWLEDGMENTS

We would like to thank the Instituto Federal do Espírito Santo (IFES) for the valuable financial support in publishing this book. Your contribution made the realization of this project possible and disseminate the knowledge presented here. We are grateful for believing in our proposal and joining us in the pursuit of academic and cultural growth. IFES played a fundamental role in this process, and your partnership is invaluable.

AGRADECIMENTOS

Agradecemos ao Instituto Federal do Espírito Santo (IFES) pelo valioso aporte financeiro na publicação deste livro. Sua contribuição tornou possível a concretização deste projeto e a disseminação do conhecimento aqui presente. Somos gratos por acreditar em nossa proposta e nos unir na busca pelo crescimento acadêmico e cultural. O IFES foi peça fundamental nesse processo e sua parceria é inestimável.

In today's digitally interconnected world, the proliferation of computational mobile devices has brought about a transformative shift in how we interact with technology. The advent of Location-Based Services (LBS) represents a groundbreaking development in the realm of computer programs. LBS programs harness the power of user spatial location information to deliver services tailored to the user's specific whereabouts. For instance, think of a mobile app that displays a map with nearby restaurants or a GPS navigation system guiding you through unfamiliar streets.

However, as convenience and innovation continue to intertwine, so do concerns about the privacy of our personal information. The very essence of LBS - utilizing your location - poses potential threats to your privacy. Unauthorized access to this sensitive information can lead to unwelcome consequences. LBS providers typically offer users some control over their privacy, allowing them to dictate who can access their location data, particularly in sensitive locations. While this approach may seem reassuring, it often falls short in providing robust protection. Crafty attackers can pose as trusted entities and exploit vulnerabilities to gain access to your location information. Additionally, your whereabouts can be inferred from your past movements or a history of places visited, further compromising your privacy.

This is where our journey begins. In this meticulously researched and expertly crafted book, we delve into the heart of this privacy conundrum. The foundation of our exploration lies in a Master's thesis in Informatics from the Libera Università di Bolzano/Bozen. Here, the author introduces a LBS middleware, underpinned by a novel approach to safeguarding user privacy.

Central to this innovative solution is the concept of user empowerment. With this middleware, users have the ability to proactively determine the probability of being tracked in a particular location. In essence, you have the power to tune the privacy protection mechanism to your precise requirements, thus thwarting potential attackers. This approach, when implemented, reshapes the landscape of location-based services by ensuring that users can partake in the benefits without the nagging fear of their privacy being compromised.

Real-world testing and assessment, complemented by the use of authentic data, solidify the effectiveness of this pioneering technique. This book not only elucidates the intricacies of the LBS middleware but also provides valuable insights into its practical application in real-world scenarios.

In a world where location-based services are omnipresent, this book

emerges as a guiding light for those who value their privacy. It offers an indispensable roadmap for individuals seeking to harness the convenience and functionality of LBS while safeguarding their personal information. Welcome to the future of privacy in Location-Based Services - a future where you remain in control.

ABSTRACT

The arrival of computational mobile devices (like cell phones and laptops) allows the development of a new kind of computer program, the Location-Based Services (LBS). This kind of program is characterized by the use of user spatial location information in order to provide services according to where the user is (e.g. a program in a cell phone that shows a map with restaurants near the user). Consequently, the availability of this kind of information to third parties can become a threat to the user's privacy, since unauthorized persons can access this information. Usually, LBS providers allow users to control their privacy by choosing who will have access to the user location when they are in sensitive locations. This approach is not good enough, since an attacker can pretend to be a trusted party and then steal this information and these locations can be inferred from the user last position or history of places visited. In this MSc thesis a LBS middleware with a new approach to this problem is presented, where the user is able to choose the probability to be found in a given place and then tune the privacy protection mechanism in order to be protected from this kind of attack. This middleware, together with real data, allow this technique to be tested and assessed in real-world situations. Consequently, this LBS middleware and this approach allow an user to utilize a Location-Based Service without fearing for his privacy.

KEYWORDS: privacy, LBS, middleware, inference attack

SUMMARY

CHAPTER 1

INTRODUCTION

The first computers date to the middle of the last century and by that time they usually occupied a enormous area and demanded highly specialized professionals to operate it. In contrast, nowadays some computers are small mobile devices that almost 50% of the world population carry in their pockets [7]. Mark Weiser define this integration of the computer into the everyday life (”everywhere, everytime computing”) as Ubiquitous Computing [37].

One of the main research fields in the Ubiquitous Computing area is the Location-Based Services (LBS). This area is characterized by the use of geographical information about the various entities inside a system, as users and objects, to provide services according to the user location. Various works had been published in the LBS area, and despite its novelty, some commercial applications were already developed and deployed. But as many new technologies, Location-Based services provide bring new facilities together with new challenges.

LBS raise the problem of protecting the users privacy, since the users should be have their locations tracked in order to provide a service. This situation become a paradox by the fact that the availability of the user location is the main need for the execution of a LBSs, but the privacy protection main goal is to control this availability!

Some facts helps to increase this problem. One of them is that some locations are more sensible them others, needing a dynamic privacy protection according to where the user is. Another fact is that some services providers may be attackers trying to break an user privacy, posing as service providers in order to get their user location. On top of that, even if the user only make his location available in few situations, his other locations can be inferred by the places that he had visit in the past, as his last know location. All these facts contribute to increase the important of the protection of the privacy of LBS users.

Another problem that interfere with the development of LBS is the complexity created with this kind of application. LBS are applications that demands complex services, as geographical content providers, and computational infrastructure, as wireless networks and multiple servers. Furthermore, these applications demands a high integration between all these services and the infrastructure, increasing even more the complexity of this kind of project. A middleware is a common strategy to this issue, fully integrating these systems in a transparent way.

These and other challenges motivate this work. The middleware developed here shall deal with all these requirements, solving these issues and integrating the infrastructure in a single and unified way. This will allow the execution of LBS in simple and safe way.

1.1 OBJECTIVES

The main objective of this work is to propose a LBS middleware with support to location privacy protection from inference attacks. In order to accomplish it, first LBS applications will be studied in order to get the basic knowledge of this area. Second, the privacy protection problem will be studied, so the main guidelines to development of the project will be traced. Then, the more technological aspect of the project will be studied, the LBS middlewares. These are fundamental steps necessary for the development of this work.

Various aspects should be taken in consideration in order to create the solution. This is due to the fact that, in order to create a LBS middleware with privacy protection from inference attacks, the technological aspects of middlewares, the sociological aspects of privacy and the mathematical aspects of inference attacks should be addressed in an integrated way. These are aspects that need to be contemplated to create the final solution.

1.2 METHODOLOGY

The methodology employed in the development of this work involve regular studies and meetings to discuss the main field of this research, middlewares for LBS with privacy protection. First, the study started on the broad area of LBS, where its characteristics were gathered and some examples were studied. This was needed in order to get the information needed as the basis to start the work. Second, the study focused on privacy protection in this type of system, studying the main mechanisms, their advantages and disadvantages. This phase allow a fine-grained choice of the privacy protection method, together with the constraints and requirements that it impose to the middleware. Next, LBS middlewares were studied, with the requirements imposed by this technology and some examples already studied. Finally, according to all the information gathered in the previous phases, the middleware was designed, stating every module and how they interact in order to execute a service. To sum up, an example of an user executing a service in the middleware with his privacy being protection was developed as a proof of concept.

1.3 ORGANIZATION

Besides this chapter, this thesis is organized in other five chapters: LocationBased Services, Privacy Protection in LBS, LBS Middlewares, A LBS Middleware with Privacy Protection from Inference Attacks and Conclusion and Future Work. The chapter 2, Location-Based Services, is an overview about this topic, presenting the definition, characteristics and some examples this kind of system. The chapter 3, Privacy Protection in LBS, focus in the privacy issues that LBS systems present, with the problem definition and the main methods used. The chapter 4, LBS Middlewares, define a LBS middleware, its requirements and present some examples of this type of system. The chapter 5, My Middleware, have a

detailed description of the middleware created in this work, with the requirements, detailed architecture with every module description, how they interact in order to execute the required services and an example of utilization. In the last chapter, Conclusion and Future Works, some conclusions are presented based on the results of this work and also some future work suggestions.

LOCATION-BASED SERVICES

This section present the field of this thesis, defining concepts necessaries for the understanding of other concepts further presented. First, the definition of Location-Based Services is presented, together with the actors involved in their interactions and how they exchange data between them. Second, in order to provide a concrete idea about this kind of service, examples of LBS are presented, some real-world applications (and others still being developed by the research community). Finally, the necessary characteristics of LBS are presented, according to necessary functionalities imposed by the examples later presented.

2.1 DEFINITION OF LOCATION-BASED SERVICES

Generally speaking, Location-Based Services are basically any kind of software that use location information about its actors to provide services to its users. Usually these services don't use only the location of an actor to provide a service, but also other related information, as nearby places or other information about an actors. An actor can be any entity, as a person or a object, which have location information available to a service provider. At the same time, a service provider is any computer-based system which use this information to perform a task for an user. Even though LBS can be a desktop program, its usually implemented as a mobile application, due to the fact that the user location information can be used in order to provide services to other users or to himself.

Despite the many years of research in this area, there is no common definition or terminology for LBS. In this thesis they are defined, according to [36], as:

LBSs are information services accessible with mobile devices through the mobile network and utilizing the ability to make use of the location of the mobile device.

Besides Location-Based Services, many others terms are used to refer to applications which use location information, as location-aware services or locationrelated service. In this thesis no distinction is done between these terms and all applications are referred as LBS.

Sometimes LBS are considered a subset of other type of application, called Context-aware Services. [18] define these applications as:

A system is context-aware if it uses context to provide relevant information and/or services to the user, where relevancy depends on the user task. and context as:

Context is any information that can be used to characterise the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves.

Context information is classified by [33] as Primary or Secondary context, if it's acquired using sensors or derived based on other context information, respectively. Primary context is further classified in time, identity, activity and location context, and Secondary context in Personal, Technical, Social, Physical and Spatial context (derived from location

context). Accordingly to these definitions, LBS are a special case of context-aware service that use only location and, consequently, spatial context information to provide services to its users.

Is important to stress the meaning of the terms position and location used in this thesis. In other types of texts they can have the same meaning, but when discussing LBS they have two different meanings. Position refers to a point where an object is in reference to another object. For example, a cellphone can be 100 meters from an antenna. Location have a more informative meaning. Location refers to a point in the space in reference to any object. For example, the Free University of Bolzano main building is located in the latitude 46.49, longitude 11.35 (of course, not exactly). Besides the cases that explicitly affirm that they have the same meaning, these are the meaning of these terms.

Another important difference between terms that must be stressed is between weak and strong privacy protection. As pointed by [22], this difference is even more important when dealing with location privacy protection in sensitive areas. In weak privacy protection, the information of an user is available only when he desire. For example, an user may want not to his boss to know his location when he is outside of his office. In addition to that, strong privacy protection ensures that the information of an user is not *inferred* from an attacker. With strong privacy protection, the same user from the last example may want his location not to be guessed by his boss. This is an important difference, especially when choosing the privacy protection method implemented in a system.

2.2 LOCATION-BASED SERVICE ACTORS

A LBS system is composed by a supply chain involving many actors performing different tasks in cooperation to execute a service. Even though the tasks performed by different actors are independent, an entity can have more than one role in the supply chain. An example is when a person use a service to find where he is. In this case, the person performs the roles of user and target of the location service. The actors focused here are only the ones involved in the operational execution of LBSs and non-operational roles, as the responsible for the standardization or vendors of the technology involved, are not considered. According to [26], the actors are the following:

Target any entity that can have his location tracked, usually in an automatic way, using a GPS, for example.

Position Originator actor that track the position fixes of the Targets using a certain positioning method. Can be performed by the actor being the Target (e.g. a person using a GPS receiver) or the operator of a networkbased positioning system (e.g. a cellphone company).

Location Provider manage the position fixes gathered from Position Originators, refining and transforming coordinate systems in order to provide high level spatial information

to LBS Providers.

LBS Provider the central actor in the LBS supply chain, which combine the spatial information received from the Location Provider and geographic information from the Content Provider in order to provide LBS to the

User.

Content Provider actor that maintain a *Geographic Information System (GIS)* with a spatial database of geographical content, as maps and routes, and provide this information to support the LBS Provider in his service.

LBS User request services to the LBS Provider, usually using a mobile device as a cellphone or PDA.

Owing to the fact that each actor perform different tasks and deals with different data, in this supply chain they cooperate exchanging different types of information between themselves:

Position Fix position of a Target according to a Position Originator, send to the Location Provider in order to create Location Data

Location Data location information in an format specified by the LBS Provider, together with other information, as target's identifier and data's quality. Is supplied by the Location Provider to the LBS Provider.

Geographic Information description of a geographic entity (e.g. a street), created by the Content Provider to the LBS Provider.

Application Data response of a LBS to the application of an user.

All the actors and the data types exchanged between them can be seen in figure 1, from [26].

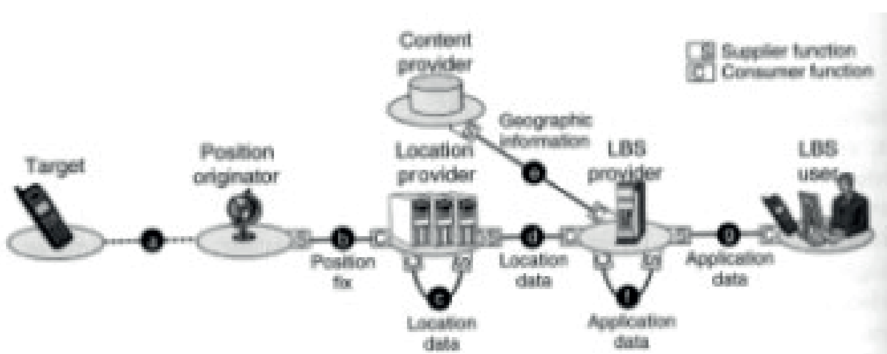


Figure 1: LBS Supply Chain.

The supply chain interaction start with the gathering of the position of a *Target* in relation to a *Position Originator*. Then, with this information, this actor can create the *Position Fix* and send it to the *Location Provider*. Now that the Location Provider know the

position of the target according to a Position Originator, it can create the *Location Data* and provide it to the *LBS Provider* in the format desired by this actor. Meanwhile, the *Content Provider* send some *Geographic Information* to the LBS Provider, usually related to the Location Data. Finally, the LBS Provider can use all this information to perform a service and send the output of this service, the *Application Data*, to the *LBS User*.

It is important to stress that more them one entity can perform more them one role in the LBS supply chain. For example, when an user provide his location information to a LBS Provider, the user perform the roles of Target, Position Originator and Location Provider. The definition of roles in the LBS supply chain is a flexible way to understand how all the entities act when executing a generic LBS.

2.3 CHARACTERISTICS OF LOCATION-BASED SERVICES

LBS can be conceptually categorized according to the kind of functionality it support, usually being independent and combined in order to perform a desired service for an user. This categorization is extracted according to some commons types of applications that can be supported by an LBS system. According to an analysis presented in [32], LBS applications can be conceptually characterized by a set of independent characteristics that can be combined in order to provide a specific LBS. A service that provide simple LBSs can have a subset of these characteristics but an general purpose LBS system should support them all. They are the following:

Push / Pull-based applications this characteristic refers to how the interactions start in the system. While in a Pull-based LBS the requests are initiated by the LBS User, in a Push-based system the infrastructure start the interactions based on the occurrence of a specific event.

Direct / Indirect Profile each user have a profile with information about the user itself (e.g. privacy preferences) and the current request (e.g. contextual information). This profile can be build directly, by asking the users in the subscription phase, or indirectly, from third-parties or by the interaction patterns.

Availability of profile information profile information can be available at request time, usually in the mobile terminal, or already available in the LBS. The first case have the advantages of a higher control over over the user information, but this characteristic do not support a selective push model and have higher request payloads.

Interaction Scenarios Since the actors involved in the service requests and responses can be either mobile or stationary, there are four cases of interactions scenarios:

Stationary requester and provider: in this case, there is no needof dynamic management of location information. Example: a user requesting a map from a personal computer.

Mobile requester and stationary provider: since only the location ofthe requester

changes, there is need of a dynamic location management only in the requester side. Example: a car using an automotive navigation system to get information about a street.

Stationary requester and mobile provider: a scenario similar to thelast one, only changing with entities are in the role of the requester and provider. Example: a car automotive navigation system service provider requesting information to a mobile terminal.

Mobile requester and provider: in this scenario, the coordinator ofthe LBS can be a central provider or a distributed system. Example of the first case: mobile user of a LBSN requesting information about another user through a server. Example of the second case: clients of an ad-hoc network requesting information about each other.

Source of Location Information location information can come from various sources, as provided by the user, by an infrastructure of sensors or by third-parties.

Accuracy of Location Information the quality of the location information demanded by the applications can range from meters to kilometers, as well as the location positioning infrastructure. These facts can constrain the kind of application that can be deployed in a LBS system.

Kind of Information Sources usually LBS do not use only location information in their applications, providing static or dynamic information, as related to the locations being used or the traffic conditions in a road, respectively.

2.4 EXAMPLES OF LOCATION-BASED SYSTEMS

The examples presented here are real-world application of LBS. They show the new opportunities created by this area, the value added by the use of location information in mobile services and that, despite the issues involving this kind of service, how useful they can become.

2.4.1 Navigation Systems

A straight forward application of LBS are the map-related systems and a popular example is Tom-tom [10], an automotive navigation system developed by the Dutch company Tomtom NV. The Tomtom is a series of products that are basically mobile devices (called *Units*) with a touch screen interface and a *Global Positioning System (GPS)* receiver. The interface shows a bird's-eye view of the road or a direct-overhead map, with functionalities to display the directions to a desired place or information regarding the places, as weather updates and traffic alerts. The GPS receiver provide the location data necessary to show in the map the places in which the user need information, as roads and cities. This application of great popularity is an example of how LBS can create worldwide business opportunities.

2.4.2 Location-Based Social Networks

The widespread of Internet access to common users allow the appearance of Social Network services, on-line communities where people can share common interests. In the same way, the adoption of mobile devices by the same type of users create a subtype of Social Networks, the Location-Based Social Network (LBSN). Likewise the first, LBSN allow the users to get in contact with other users, share media and interact using programs, as well as perform tasks using the location information of themselves or other users. They can locate friends, share photos embedded with location information or find nearby people with the same interests that he have.

A example of LBSN is Loopt [8]. Its described as:

Loopt shows users where friends are located and what they are doing via detailed, interactive maps on their mobile phones. Loopt helps friends connect on the fly and navigate their social lives by orienting them to people, places and events. Users can also share location updates, geo-tagged photos and comments with friends in their mobile address book or on on-line social networks, communities and blogs. Loopt was designed with user privacy at its core and offers a variety of effective and intuitive privacy controls.

Its basically allows users to connect to friend through their mobile phones, give suggestions and share information about places, and explore content created by other users. An example of use of Loopt, from [35], can be seen in the figure 2.

2.4.3 Emergency Services

Sometimes victims of accidents or disasters don't know where they are or are unable to transmit this information to rescue teams, in case in which the victim is lost or unconscious. Emergency services is a type of LBS application where is necessary to discover the location of a victim and execute services that need this kind of information. A good example of organization which provide this kind of service is the COSPAS-SARSAT [3] (COSPAS is an acronym for the Russian words "Cosmicheskaya Sistyema Poiska Avariynich Sudov" which mean "Space System for the Search of Vessels in Distress" and SARSAT is an acronym for Search and Rescue Satellite-Aided Tracking):

an international, humanitarian satellite-based search and rescue system that has helped save over 20,000 lives worldwide since its inception in 1982 (total as of June 2005).

It is supported by various countries, including Canada, France, Russia and United States, and works 24 hours a day, 365 days a year, for free to the beacon operators. In the case of a emergency (figure 3, from [3]), emergency beacons are activated, which transmit signals through satellites and ground stations until a rescue center receive the necessary information. At the end, the rescues teams can perform the necessary tasks to help the victims, supported by the available information gathered using the system.



Figure 2: Loopt - a Location-Based Social Network.

PRIVACY PROTECTION IN LBS

This section present the main subject of this thesis, the privacy protection in Location-Based Services. First we define the general privacy concept and them focus on the definition in LBS. Next, some commentaries are done regarding the characteristics of LBS and how they influence the privacy protection in these systems. Finally, the main privacy protection mechanism are presented, with their problems and advantages.

3.1 PRIVACY DEFINITION

Privacy is such a common and ubiquitous concept in daily lives that seens not to have a single definition, depending on the context and in which historical moment it is used. The concept started to appear in the societies with distinction between public and private domains, as in the ancient Greece and China, where it started to became a important social concept. In the 14 century, start to appear the first concepts of modern privacy with the *Justices of the Peace Act*, where voyeurs and eavesdroppers were arrested, and these concepts where evolving with the development of new technologies, as photographic cameras and telephones. In the later years, with the evolution of computational and communication devices, the issue of electronic privacy start to grow in atten-tion and modern laws dealing with this issue started to appear, as the Directive 95/46/ec [19] in the European Union.

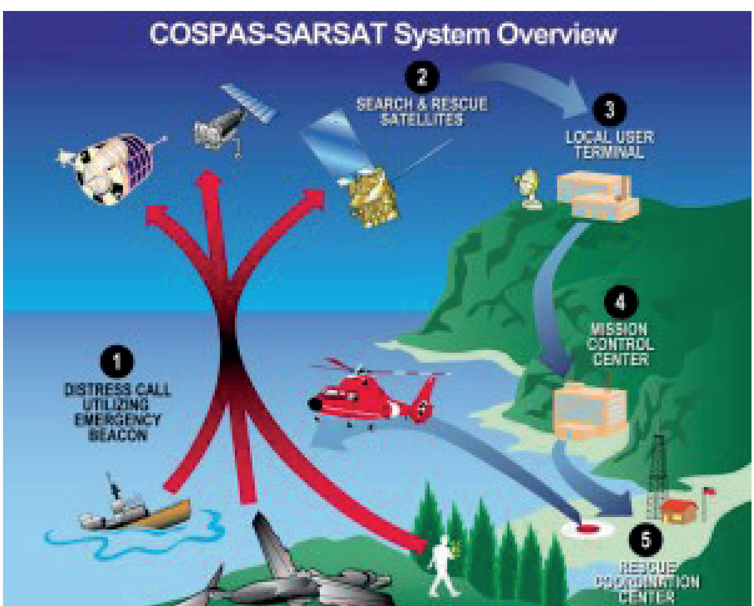


Figure 3: COSPA-SARSAT System Overview.

In our actual society, the definition of privacy, by [12], more often accepted and cited is:

Privacy is the claim of individuals, groups and institutions to determine for themselves, when, how, and to what extent information about them is communicated to others.

The definition also state four characteristics needed in order to have privacy:

Anonymity: interact with others entities without being identifiable.

Solitude: be able to be alone, without intrusions, interruptions and observations.

Intimacy: decide the context (with who, when and how much) that theinteraction will occur.

Reserve: decide when to make available an information.

When applying this definition in LBS, the privacy concept is defined by [15] as:

The ability to prevent other parties from learning one's current and past location.

This definition lacks on state that, according to the definition made by [12], privacy is not only the ability to prevent access to a location information (since an entity can allow other entities to access this information), but to control this access. In addition, since an user future location can be inferred with the past locations, it's also necessary to stress, adding meaning to the definition, that privacy in LBS deals not only with the control of the access of past and current locations, but also of future locations. Consequently, the new definition become

Privacy in Location-Based Services is the ability of entities to determine by themselves who in what context will have access to their past, present and future locations.

Besides the definition, the four characteristics are also adapted to LBS:

Anonymity in LBS: use LBS using a pseudonym, without providing thetrue identity.

Solitude in LBS: be able to control the interactions with a service bysubscribing and unsubscribing from it.

Intimacy in LBS: the user should control for who and in which situationhis location can be available to others users. It's important to stress that this control should be available not only to direct requisition of location information by actors, but also to third-parties that can have access to it. This is specially important in the case of an inference attack, where non-authorized third-parties try to guess and access the user location.

Reserve in LBS: each position send to other actors should be explicitlyauthorized.

3.2 PRIVACY PROTECTION IN LOCATION-BASED SERVICES

Even though LBS can provide innovative services with useful functionalities to its users, the possibility to loose their privacy can make users avoid these services. The advent of digital fraud made the IT services acceptance be limited to the quality of the security and privacy mechanisms provided by the service, specially in the case of important data. Given the sensitive nature of location and the other informations can be associated to it, a solid

privacy protection mechanism can encourage users to adopt LBS in their daily lives.

Moreover, compared with other IT services, LBS have some characteristics which make privacy protection even more challenging. First is the fact that, together with other information as address and age, location belongs to a category of high-level information that usually users will to be kept private. This information can be used to create evidences of delicate life styles, as when belonging to political or religious groups. Second, as showed in the section 2.2, in order to perform a service, the location of an user should be transferred through various actors, usually in a transparently way. This increase the risk of misuse of the information by the actors or the user be not aware of who can access his data. Finally, the fact that a LBS can automatically track the user during his everyday activities, even when he is not aware that is using the service, can break the privacy in a 24/365 manner.

Moreover, as noticed by [26], location privacy protection in LBS is a dilemma. The sharing of location information by different actors is a key factor to enable LBSs, whereas location privacy protection is the allowance or denial of this sharing, depend of the case. In some cases, the continuous tracking of location is mandatory for the operation of a service. Cellphone companies, for example, should keep track of their users in order to create the connections necessary for the phone calls. The point is to protect the users privacy and, at the same time, allow the service to operate.

3.3 PRIVACY PROTECTION MECHANISMS

Given the functionalities that LBS provide to its users and the possible threats that can happen to their privacy, a lot of research has been performed in privacy protection mechanisms in this area. This section present the main methods, with their approaches and problems, but since this is a important issue further research is still need to investigate them.

3.3.1 Privacy Policies

Privacy preferences cannot be created according only to a location sensitivity level or the service being invoked, but they are mainly influenced by personal ideas and inclinations. This allows the LBS to publish his constraints that are used by the client device to check whatever the user should subscribe to the server or not. Privacy policies are a way to specify the rules of release of the location information of an user according to his preferences. According to [31]:

Privacy policy is an assertion that a certain amount of information (identity or identifier plus location) may be released to a certain entity (or group of entities) under a certain set of constraints

Examples are:

- My boss is allowed to get my location, while I'm inside of my office building.

- My friends are only allowed to get my location after I explicitly allow.
- Colleagues can get my location, but with an accuracy of 10 km^2.

There are numerous types of constraints of a privacy policy and they can vary according to the characteristics of a LBS. For example, in a LBSN is desirable for an user to be able to choose the other users that can have access to his location, while in an location-based emergency service the user may allow his location to be available only in case of emergencies, no matter who will access his location. The constraints should be chosen according to the characteristics of the LBS. According to [28], the possible constraints can be:

- Actors involved: set of actors involved in the execution of a LBS. Example of actors can the other users or LBS providers.
- User confirmation: while is desirable to the LBS to work in a transparent way, without prompting the user too many times, the user can be notified and asked for confirmation if this location can be available to some services.
- User data and context: some data from the user, together with contextual information, can be used to constrain the release of location information.
- External services: third-party validation services can be used in order to validate if a location can be send to a LBS.
- Statement: a policy statement can be created to validate each type of request from each LBS, providing fine-grained privacy protection.
- Limit time: the user can constrain his location to be available depending of the time (e.g. during working hours)
- Limit location: similar to the constraint, but the user can restrict his location to be available depending where he is.
- Quality of service/accuracy: the user can limit the accuracy of his location available to a given LBS.
- Anonymity: services which don't need identity information from the user can be used in an anonymous way, by using pseudonyms.

Even though privacy policies are used in many LBS architectures (CITE!), they have some restrictions and drawbacks that should be considered. In order to be used in a proper way, the privacy policies should be machine readable, allowing the creation of automatic validators for the policies. Besides that, the software architecture should have mechanisms to ensure that the services are dealing with the data in the way that they specify in their policies. [16] present one of these mechanisms, that is keep a log with all the requests being made in the architecture, basically recording who got access to what in which situation. This is due to the fact that privacy policies are based on trust. The user trust that the LBS will follow his privacy policy and treat his data according to his constraints. According to this fact, is desirable that other mechanism of privacy protection should be used at the same time with privacy policies.

3.3.2 Identifier Abstraction

In this method, in order to protect the target true identity, the identifier of a target is changed to a pseudonym, in a permanent or temporary basis. For this reason, this method is not suitable for services that need the user identification, as name or other identification information. In the first way, the user subscribe to a service and them receive a pseudonym that will be valid until he unsubscribe. While this method have the benefit that the user remain identifiable for different LBS during various sessions, it also have the disadvantage that the location can be inferred if an attacker have some background information about the target. Therefore, it provides weak privacy protection.

Do deal with this drawback, pseudonyms can be temporarily assigned to a user. In this way, the user subscribe to a service and them receive a pseudonym that will be changed from time to time. This method don't have the advantage of the later way, where an user just need to receive his pseudonym once, but this fact make it harder to suffer from inference attacks. Nevertheless, if the spatial or temporal resolution of the location is high, an attacker can still break an user privacy linking old and new pseudonyms. This can be avoided by the method proposed by [15], the Mix Zones.

In the Mix Zones method, the locations are classified in application or mix zones. Application zones are places where the location of an user can be available to requesters, and mix zones are places where, each time an user enters, his pseudonym will be changed with others users in the same mix zone. This decreases the possibility to link the pseudonym of users before and after they pass through a mix zone, providing strong privacy protection.

Even though the Mix Zones method try solve the problem of linking pseudonyms, this problem can still arise if the mix zones are not carefully dimensioned. If they are too small, they may have few users (or even only the actual user), increasing the probability of linking old and new pseudonyms. If they are too large, they can have the accuracy lower them the demanded by the LBS, becoming impossible to provide the service in a proper way.

3.3.3 Disclosure-Control Algorithms

Disclosure-Control algorithms are basically methods to suppress location updates when the user is in a sensitive area. This kind of algorithms is specially important in continuous location-tracking applications, since the user can be willing to be located in any area, but not in sensitive ones. A good characteristic of this kind of algorithm is that it provides a strong privacy protection, since they can deny an attacker to infer where the user is based on past or further locations updates. [22] present three algorithms: Base, Bounded-rate and K-area.

The Base algorithm is the simplest and basically releases only location updates in areas classified non-sensitive. For each location update, the algorithm check if the actual location is inside of a sensitive area and only send the location to the requester in negative

case.

The Bounded-Rate algorithms is similar to the Base algorithm, with the addition that is ensures that location updates are not sent with a frequency higher than a predefined threshold. This lower the amount of location information released in non-sensitive areas, becoming more difficult to an attacker to infer the position of an user in a sensitive area.

Finally, the K-area algorithm, as the Base, allow location updates only in non-sensitive areas and, in addition, location updates are allowed only when they do not inform which of at least k distinct sensitive areas the user passed. A distinct sensitive area is an sensitive area that can be reached from at least one non-sensitive area and from which no other sensitive area can be reached without passing through a non-sensitive area. This avoid a location update to inform which of the k sensitive areas the user pass through.

In [22] is presented the evaluation of the algorithms in a simulation of a city composed by buildings and streets (sensitive and non-sensitive, respectively). The Bounded-Rate algorithms was configured with location updates of one, five and fifteen minutes intervals, and the K-area was configured with four and twelve houses. The Base algorithm and Bounded-Rate with one minute of interval offered almost no protection. The Bounded-rate configured with five and fifteen minutes offered between 20 and 45 % of protection, but at the same time, blocked between 50 and 75 % of location updates. This high percentage of location updates blockage can constrain the LBS used with this mechanism. Even though the K-area present protection rate of 60 to 80 % with location updates blockage of less than 15 %, it still dependent of how the the areas are partitioned. Also, it delays the location-updates when in sensitive areas, making impossible to use the LBS in this situation. On top of that is the fact that the algorithm don’t take into consideration the landscape characteristics.

The study took into consideration that

From an adversary’s perspective, a user is equally likely to enter in each building.

This fact is a major disadvantage if an attacker is aware of the landscape characteristics.

3.3.4 Information Content Abstraction

A way to protect privacy in LBS is through abstracting the information resolution, in the space or time dimension, in such a way that it becomes indistinct from from other users in the same area. This is the basic logic of the K-anonimity algorithm, proposed by [23]. In this algorithm, the location of an user is not given in exact coordinates, but by presenting an area (called cloak) and the timestamps in which the target was there. At the same time, at least k users should be in the same area within the same timestamps. This allows the user to specify the minimum value of k (called kmin) that will constrain the probability of his location to be inferred. As larger the number k is, as more anonymous an user will be, and will be harder to differentiate the location of one user to another in the same area.

The drawbacks of this approach are related to the trust relation with the location provider, the location accuracy demanded by the LBS. First, since is necessary to know the location of nearby users in order to calculate the cloak, the user should send his exactly location to the location provider, that will gather the location of other users and define the cloak area. This demands a trust relationship between the user and the location provider. Second, this contradict with the efforts to create high-accuracy location methods, since this method decrease the location accuracy. This problem can increase in areas with low population density, since it demand that k-1 users to be in the same area of the cloak. Finally, this approach don't take in consideration that some areas have a more sensitive nature them others, depending on the users perspectives.

3.3.5 Landscape-aware Methods

The Landscape-aware method is a special type of Information Content Abstraction, where the location information is abstracted taking in consideration the constraints and characteristics imposed by the landscape where the user is. The fact that, as pointed by [20] and [13], a landscape where the user is located can be not neutral and increase the probability that the user is located in some specific parts (as by barriers that can constrain the user movement or places where is more common to the user to be found), can be used by an attacker in order to infer where the user is. Another fact is that, even in landscapes where the user have a uniform location probability, some places can be more or less sensitive, regarding the user privacy. (All these facts should be taken in consideration when creating a method for protecting the location-privacy of an user.) Despite the other methods that don't take this information in consideration, the method presented by [20] use this fact to create a enhanced location privacy protection method.

This method have a Game-theoretical view of abstracting the location (called cloaking), where the actors (referred as Alice, Bob and Charlie) have a rational behaviour but different objectives. The first actor, Alice, have a tracking device that sends her location to Bob, in which she trust. Bob perform the location estimation of Alice, obfuscate the location and send it to the LBS provider Charlie, which could try to de-obsfucate the location, performing a inference attack. The work of [20] model and solve this problem as a two-player, zerosum, matrix game, finding the equilibria in which the user Alice can adjust her clock in a way to minimize the possibilities of an attacker (Charlie in this case) to infer her location, according to the sensibility of the place where she is. This allow the user to have strong location-privacy protection tailored to the characteristics of the landscape and the sensibility of the locations where they want to be protected.

LBS MIDDLEWARES

This section present a study on LBS middlewares. First, this kind of system is defined comparing it to “regular” middlewares. Second, the necessary requirements for LBS are presented and how they can influence the design of a LBS middleware. Finally, examples of middlewares are presented, together with their characteristics and main ideas.

4.1 DEFINITION OF LOCATION BASED MIDDLEWARES

A middleware is a distributed computer software that connects components and applications using standardized APIs, protocols and infrastructure services. It provides interoperability and support an easier and faster development of distributed applications, since programmers do not have to deal with the complex problems of distributed systems, as remote methods invocation.

A LBS middleware have the same characteristics as the regular middlewares, but additionally extend its capabilities in order to support LBS applications. It should spread over the entire LBS supply chain, incorporating the different infrastructures and protocols used by all the actors and hiding the heterogeneity from the LBS applications. A LBS middleware also have to deals with management aspects, as control over the quality of location data, protection of users privacy and accounting of the services usage. Any project of LBS middleware should deal with these terms. Owing to that, a LBS middleware project can reuse a regular middleware and extend it with functionalities to support LBS applications.

In [26] a conceptual view of a middleware project is presented. This conceptual view helps to understand the various components, and the functions belonging to each one, that a middleware architecture should have. As show in the picture 4, the middleware is organized in layers, between the LBS client applications in one side and the positioning methods and geographic content in the other. Firstly, the applications are the softwares (usually running on mobile devices) that uses the functionalities supported by the middleware in order to perform some service to its users. Secondly, there are three components in the middleware: Core Services, Management Services and Location Services. The Core Services have the main functions for processing location information, as navigation, geocoding and point-of-interest search. The Management Services support and control the execution of the client applications and Core Services, mapping their needs onto other sub-services. Finally, the Location Services component uses protocols as WAP and Parlay to provide location for the other components. Since this model is just a conceptual model it doesn’t provide a concrete specification, just an overview of a LBS middleware

4.2 LBS MIDDLEWARE REQUIREMENTS

LBS middlewares inherity the requirements of regular middlewares and also should deal with the characteristics of LBS applications ,presented in the section 2.3. Consequently, the number of requirements imposed on LBS middlewares is high, specially if the middleware is designed to support a wide range of LBS

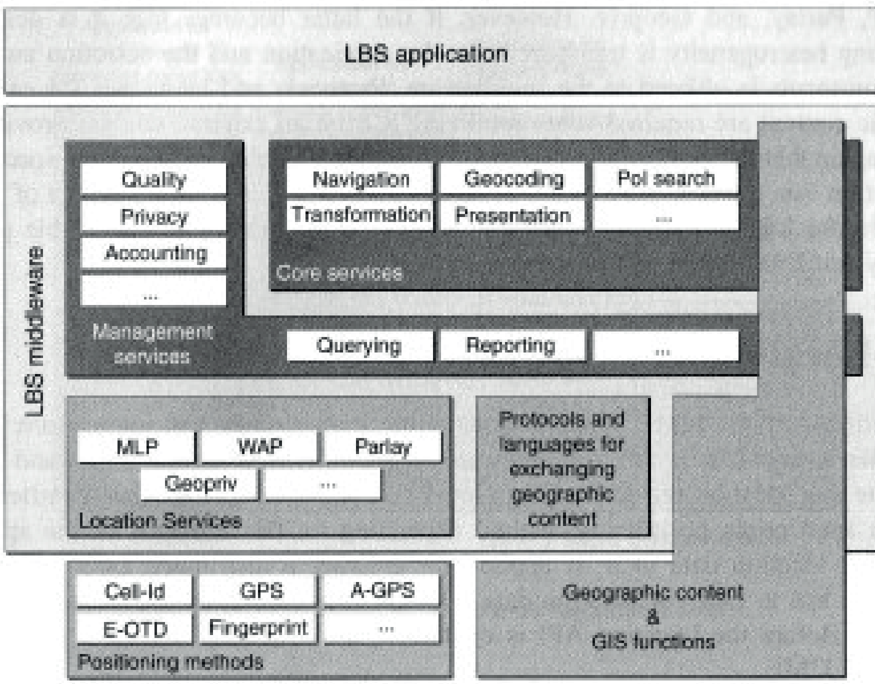


Figure 4: LBS middleware conceptual model

applications. If a middleware have a small set of desired LBS applications, some of these requirements can be avoided (e.g. a middleware that uses only one location-discovery method). According to a study present in [17] and [24], LBS middlewares should support the following requirements:

- Disconnected operations: due to the fact that the quality and availability of ne-twork connections can vary through time.
- Mobility awareness: a basic requirement inherent from the LBS characteristics, i.e., the dynamic location of the targets.
- Changes in the network topology: the network configuration can change while users move or change their devices, varying from a stable wired connection to very dynamic ad-hoc networks.
- High number of LBS users and providers, with constant profiles updates and performing operations in parallel: the high numbers of subscribers and service providers should not decrease the quality of the available services, and their

requests should execute concurrently.

- Manage content in diverse formats: the data used by users and service providers can be modelled from simple ASCII data to data organized in complex semantic ways.
- Heterogeneous notification channels: the channels used to notify users can range from simple instant messages to complex SOAP messages
- Approximate subscriptions and events: this increase the flexibility of the system, with a expressive subscription language with support to approximate data requests (i.e. fire events using approximate location)
- High availability: the communication between the nodes in the architecture (i.e. users and services) can fail in some cases, so the architecture should guarantee the delivery of messages, if this is possible.
- Accounting functions: an accounting mechanism should be used to manage the users and services, being necessary for charging and control over the architecture resources.
- Security functions: perform authentication of users and service providers, and secure distribution of content.
- Privacy for users: select the service providers which can have access to his locations, together with the granularity of these informations.
- Limitations of mobile devices: despite the fast evolution of mobile devices, the functionalities provided by the middleware should take into consideration the limitations of this kind of device, as low processing and battery power.

Accordingly to this list, the requirements not only regards the middleware architecture itself, but also the chosen communication technology, mobile devices and types of LBS applications. Since these characteristics can change a lot, also the requirements importance will vary according to the functionalities desired to be performed by the system.

4.3 EXAMPLES OF LBS MIDDLEWARES

This section presents some developed LBS middlewares as well as some research projects being developed in this area. Despite the many years of research in this area, there are few LBS middlewares available in the market, and the majority of them are focused in specific services. Other projects are not focused in the implementation of this kind of technology, but in creating standards to allow the interoperability between LBS middlewares. All this is show in the following.

4.3.1 Open LBS Middleware Platform

The Open LBS Middleware Platform (OLMP) [25] was proposed by the Telematics & USN Research Division as an architecture to guarantee interoperability and real-time

processing. These requirements were focused specially in the supporting of different mobile clients and the processing of large moving objects, respectively. The architecture is formed by three sub-systems: a set of Open LBS Components, a Mobile Gateway and a Main Memory DBMS. As a proof of concept, it was implemented as a prototype and experimented in South Korea.

The Open LBS Components sub-system use Web Services to allow applications to communicate between themselves, guaranteeing the interoperability and the combination of their functionalities independent of the platform being used. This also allow the sub-systems to be deployed in different servers, decreasing the problem of server overloading. The sub-systems are the travel advisory, routing, presentation, location utility, directory, tracking and positioning component. They are not explained in details, since their name self-explain their functionalities.

The mobile gateway is a XML parser for the data exchanged between the servers and the mobile devices. This is necessary owing to the fact that some mobile devices doesn't have the capability to interpreter XML data. This is performed in the following way: first the protocol used is analysed and the request is classified according to each service type. Next, the server interprets the request type and transform it into XML request data. Finally, this data is send to a proxy according to the desired Open LBS Component sub-system (e.g. Travel Advisory proxy) and the service is performed.

The Main Memory DBMS is used to manage all the data necessary by the other components, as GIS data. The choice of using a Main Memory DBMS is due to the fact that traditional databases systems have two major problems in managing moving objects: the disability to cope with high-update operations of moving objects and they don't support moving objects natively. Using this kind of database system, the platform allow developers to deal with this kind of information through the native functionalities supported.

4.3.2 LBS based on Java

One of the main challenges faced by LBS developers is the fact that their application should run in a great variety of mobile devices, with a wide spectrum of hardware and software configurations. Due to this fact, developing the software in Java is a natural choice of strategy to deal with this challenge. This technology allow the software to be written once and deployed in various devices. In addition, the use of Web Services make the communication easier between computers in different networks. This is the strategy used and presented by [38].

Compared to the LBS supply chain seen in 2.2, the architecture uses a restricted version of if. The mobile device sends its location information to the server in which reply with information about nearby places. The clients and servers are organized in a three-tier architecture: the first tier is composed by the software running in the mobile device,

as a positioning and browser agent; in the second tier run the Web Services necessary to communicate with the mobile devices and the services necessary by the LBS, as mapping and routing; in the third tier is the database side, mainly managing spatial data, as maps and information related to places.

The architecture procedure to invoke a service is similar to the one adopted by OMLP, previously described. The user communicate with the architecture using a mobile device through a Web Service using Simple Object Access Protocol (SOAP), indicating his actual position acquired by a GPS receiver. The server identifies the service being invoked and acts it forwarding the user location to this service. The service queries the database according to the needs of the service and use this information to perform the desired action. The result is them forwarded to the mobile device that displays the information to the user.

The results indicate that the technology chosen (software development with Java and communication through Web Services) have various advantages. The Java 2 Micro Edition (J2ME) is supported by various devices, providing portability and a very useful Application Programming Interface (API) to develop the softwares. A special feature of this language is the support for location gathering and related information, as *Point of Interest (POI)* classes. Java on the server side (J2EE) also provide the necessary portability and API, also supporting other technology, as *Enterprise Java Beans (EJB)* to separate the platform's logic. In addition, Web Services provides communication in a transparent and interoperable machine-to-machine interaction over a network.

A LBS MIDDLEWARE WITH PRIVACY PROTECTION FROM INFERENCE ATTACKS

This section present the main result of this work, a LBS middleware with privacy protection from inference attacks. First, is present the requirements necessary by regular LBS middleware (discussed in details in the section 4.2) together with the peculiar requirements imposed by the protection from inference attacks. In addition, is present also the characteristics of this LBS as well as the reasons for these characteristics. Second, the architecture of the LBS is present, its modules and how they interact, with special attention to the module responsible for the privacy protection mechanism. Finally, is presented an example of the architecture in operation in a real world scenario, showing how it succeeds to protect the privacy of its users.

5.1 MIDDLEWARE REQUIREMENTS

As mentioned in the section 4.2, the requirements of a LBS middleware can change in importance according to the functionalities provided by the middleware. Since the middleware developed in this work intend to be as much general as possible, and at the same time provide protection from inference attacks to the location privacy of its users, the requirements will be linked to specific parts of the architecture, as the modules and the communication technology used between them. Also, not all the modules will be described in details, only the ones necessary to allow the execution of LBS.

To begin with, to support disconnected operations the communication between the users, service providers and architecture is performed in a asynchronous manner using Web Services. The service requests done with the users subscriptions to the platform will not lock the execution of others tasks in the users devices. In the same way, the service providers will receive the service requests. The only peers involved in the system that should always be on-line are the modules of the architecture itself, since they provide its basic functionalities.

The asynchronous Web Services also solve the problem of high availability of the architecture. The asynchronous messages exchanged by the platform, its users and services guarantees that they are interchanged when the nodes are available for communication. In addition, the platform being composed by several modules communicating through Web Services allow them to run on different computers at the same time. This characteristic make it easier the high availability to be implemented.

The adoption of Web Services technologies also fulfil the requirements of support changes in the network topology as well as the problem of heterogeneous notification channels. Since this technology abstract the lower levels in the protocol stack, any kind of network topology can be used, as long as the current topology supports Web Services.

Consequently, the changes in the network configuration will be transparent to the users and service providers. This is a minor problems owing to the fact that Web Services use technologies that became standards with the widespread of the Internet. In the same way, any type of notification channel can be used as long as it can be send through Web Services. This characteristic decrease the complexity of the architecture but make some notifications channels, like SMS messages, harder to implement.

The use of Web Services technology allow a standardized communication between the various actors involved in the architecture, fulfilling various requirements. Even though the Web Services is a relative new technology, currently its supported by a high number of devices, even having APIs to allow it to be used by mobile services [2], taking into considerations the limitation of mobile devices, another important requirement for the middleware.

Similarly, in order to avoid the limitations of the mobile devices, the most time and energy consuming processing are done in the architecture. Even though these devices had evolve from simple cellphones to devices with capabilities found before only in personal computers, as WIFI connections and high storage devices, energy and processing power are still important issues when compared to regular computers. Almost all the necessary processing for the execution of the services (as the dealing with geographic content, users and service providers management, privacy protection, etc) is done in the architecture. The mobile devices should basically invoke the services and consume the services response data through the architecture.

In order to manage the different contents and the formats that they are available, the module Content Provider Proxy will act as an abstraction layer, providing the content in a single and transparent way to the modules and service providers. Instead of a module or service provider request the content directly to one of the content providers, they will request the content to the Content Provider Proxy. All the data is modelled in a standardized way, transforming the data in the format desired by the users/service providers from/to the one provided by the Geographic Content providers. This module will handle the request, gathering the content in which fits more from the Content Providers and returning the content in the desired format to the requester.

The middleware have a special module to protect the privacy of its users, the Privacy Protection Reasoner module. This module is responsible to define the granularity of the location of its users that shall be available to the service providers. In order to accomplish this, the module receive the level of privacy protection desired by an user, the landscape data of the users actual location and compute a cloak, i.e. an area large enough to cover where the user is in which his privacy preferences are satisfied. This allows the execution of the service without any privacy problems for the user.

Since in this middleware the user himself provide his location, abstracting the way that the positions and locations are gathered, the requirement of ”mobility awarness” is

automatically satisfied. Even though this do not match some real world situations, like when the location is gathered from a cellphone network, it's enough to as a proof of concept of the functionalities of the architecture and this situation can be accomplished in the real world, as if the user have a device with a GPS or other location tracking mechanism.

Owing to the fact that the middleware is designed based on modules that interact between themselves, they can run in parallel on different servers, enabling a multitude of LBS users and providers, constantly updating their profiles and performing operations in parallel. Since this requirement depends a lot on how the implementation of the middleware is done, is necessary experimental evaluations in order to assure that the requirement is fulfilled. After this evaluation, some modifications can be done in order to allow the middleware to support a high numbers of subscribers and service providers without decreasing the quality of the available services.

The middleware have a module, Content Provider Proxy, that deal with the complexity of the geographical data needed by LBS. The module, using a standardized and rich language, allow flexible data requests. This makes possible the gathering of simple geographical information, as a point in the space, or complex, as streets and buildings. Also, the LBS usually execute their services not with the exact location of the user, but with an area where the user is according to his privacy preferences. This permits the execution of the service, but protecting the user privacy.

The module Subscription Manager, together with the User Profile Repository, manage users and services. The User Profile Manager keep all the data, as login and billing information, necessary to deal with the accounting of the users. At the same time, the Subscription Manager manage a log with all the interactions with the services providers, and control the execution of the other modules of the architecture. This mechanism permit this information to be used on business and to control the use of the middleware.

There are two main points in the security functions required by the middleware: secure managing of data and secure communication. The first point relates with the data belonging to the users, service providers and geographical content. The module Subscription Manager deals with the authentication and access control of users and service providers, safely storing their data in the User Profile Module and in the Subscription Manager itself, respectively. In the same way, the module Content Provider Proxy safely manage the content provided by it. Meanwhile, the communications are all performed using secure channels, implemented as Web Services. Owing to the fact that security in Web Services and LBS is such complex area, this topic is not focused here, creating opportunity from further research.

In conclusion, the middleware requirements are all fulfilled taking in considerations the architecture goals and characteristics. The main characteristic focused in this work is the privacy protection mechanism and the requirements imposed by it. Is important to stress that some questions are still open, in which create opportunities for further research but impose problems for the wide spread of LBS. The table 1 summarize the requirements and

how they are fulfilled by which modules and technologies adopted.

Disconnected operations	asynchronous Web Services
High availability	Web Services, architecture composed by modules
Changes in the network topology	any topology that supports Web Services
Heterogeneous communication channels	restricted to notification channels that can be deployed over Web Services
Limitations of mobile devices	Web Services, main processing in the architecture
Manage content in diverse formats	Content Provider Proxy module provide the content in a standard transparent way
Privacy for users	Privacy Protection reasoner module guarantee a high protection of privacy
Mobility awareness	automatically satisfied since the user deals with the location gathering
High number of users and services	architecture composed by modules, but a practical evaluation is needed
Approximate subscriptions and events	Content Provider Proxy have a flexible and standardized language for requests
Accounting functions	the modules Subscription Manager and User Profile Repository manage the services and users, respectively
Security functions	the modules Subscription Manager, User Profile Repository and Content Provider Proxy safely store all their data, and the communications are performed using secure Web Services

Table 1: Requirements fulfilment summary

5.2 MY MIDDLEWARE CHARACTERISTICS

To provide a conceptualization of the middleware developed in this work, its necessary the analysis of its characteristics according to the ones discussed in the section 2.3. This define the types of applications supported by the middleware, i.e. the ones which need the characteristics supported. Even though the middleware in this project tries to be a general-purpose middleware, some restrictions should be made according to the constraints imposed by the project, in details in the following.

The system basic interaction type support only the Push-based applications, i.e. the ones where the user initiate the requests. There are further plans to also allow Pull-based interactions, and the architecture was designed in order to allow an easy adaptation to it, but they are let for further research.

The users have direct profiles build during the subscription phase. Even though this option can raise some privacy issues, since the middleware acts as a trusted party in the system the privacy problems do not occur. It keep any information necessary to maintain the interaction with the architecture, such privacy preferences and billing information, but the only information focused here are the first ones.

The profile information also is available at the architecture, gathered during the

subscription and kept in user profile. This approach have the disadvantage of a lower control over the user information, but since this middleware acts as a trusted party, this issue is overcomed. This allow a fine-grained control over the user data and have lower requests payloads, helping to avoid the mobile devices limitations.

According to the categorization presented in the section 2.3, the possible interaction scenarios vary according to the nature of the necessary location information, dynamic or static. In this architecture, the users provide their location information, no mattering the way that the location is obtained. Due to this fact, the location information is gathered only when its necessary, so the location information can be gathered in a dynamic or static way. Therefore, the middleware support the four interactions scenarios (stationary requester and provider; mobile requester and stationary provider; stationary requester and mobile provider; stationary requester and mobile provider).

All the location information come from the user in a transparent way to the architecture. This can be done through location receivers connected in the user device (as GPS receivers) or provided by a third-party service. This keep the logic of the architecture simpler and moves the responsability of the gathering of location information and the managing of its quality to the users, but limit the way that the user gather his location. For detailed information about positioning and location methods, consult [4].

In conclusion, usually LBS do not use only location information to provide services, but also information related to the actual position of the users, as nearby points-of-interest, etc. This middleware have a module, the Content Manager Proxy, dedicate to deal with this aspects. It deals with the complexity and functionalities of the different services available, avoiding possible limitations for the service providers. This module is detailed n the further sections.

5.3 MY MIDDLEWARE ARCHITECTURE

The middleware design was made taking into considerations the characteristics of LBS, focussing in the mechanism to protect the privacy of its users and the requirements imposed by this technology. Even though the aim of the middleware is be as general as possible, regarding the types of LBS applications supported, some restrictions were made in order to keep it consistent with the requirements while keeping it functional and simple.

The middleware is basically an architecture composed by modules which interact using Web Services in order to execute services requested by users and provided by service providers. Provided that this system is a middleware, the users and service providers interact with the architecture through an API, available as Web Services in this case. The overall architecture is showed in the picture 5 and the main components are described in the following sections.

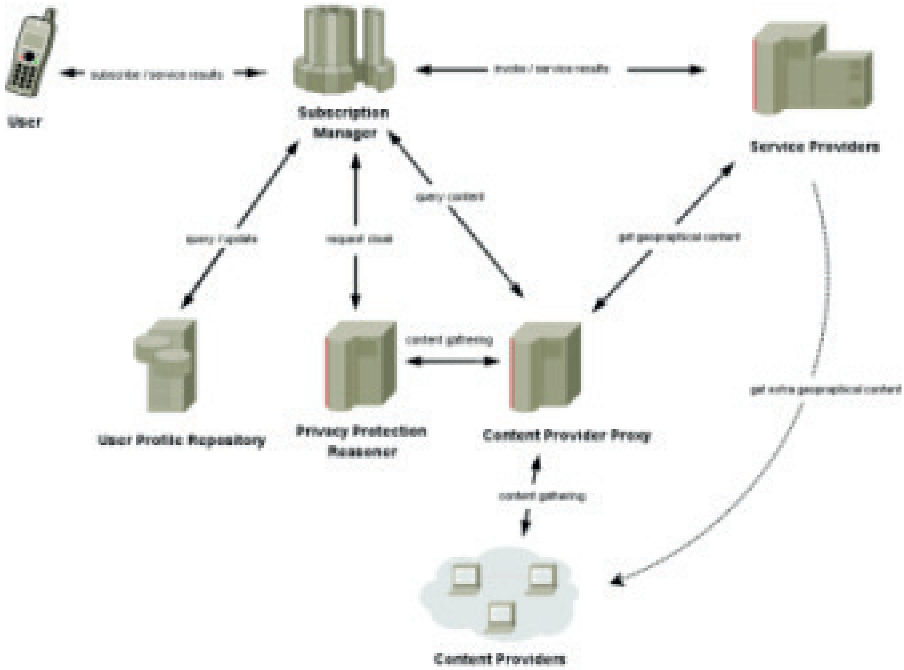


Figure 5: LBS overall architecture.

The modules are described in a concise way, showing their main functionalities and allowing them to be further extended. Therefore, some modules are more focused than others, but all the main characteristics are outlined. They were designed according to the requirements presented before and a summary showing the fulfilment of each requirement is presented in the end of the section.

5.3.1 User

The user represent any person (or other entity) willing to execute one service, but in the same time not disposed to have his privacy threaten. Since its only possible to interact with a middleware using a computational device, in this case a mobile device, the user and the mobile devices are treated as one single entity, but named in different ways in two different circumstances: called user, when dealing with a person that have his personal data and want his privacy to be protected; called mobile devices, when dealing with a computational device with capabilities to communicate electronically with the middleware. In this section, since only the aspects of communication are discussed, the term user apply for the mobile device. Furthermore, this entity is not really situated inside in the architecture, but since it uses the middleware services it is described here in order to help to understand his interaction with the middleware.

An user interact with the middleware through the use of an API available as Web

Services. To start the interaction, the user send a subscription to the module Subscription Manager, informing his location and the service that he wants to invoke. Transparently to the user, the middleware execute the subscription and return the service results to the user. Owing to the fact that this communication is done asynchronously, the user can perform other actions while waiting for the answer of the subscription.

The result send to the user is treated in a general way, allowing it to vary and be suited to a wide range of LBS applications. This is necessary, as seen in the section 2.4.

5.3.2 Subscription Manager

The Subscription Manager is the central component of the architecture, responsible for treating the subscriptions sent by the users and perform the necessary tasks to execute these subscriptions. Different from other modules that have capabilities related to specific tasks, as protect the privacy of the users or provide geographic content, the Subscription Manager main function is to invoke the others modules in the proper order to execute a subscription. In addition to this task, the module also deals with functions more related to the managing of the architecture, as accounting and security.

The Subscription Manager perform a series of steps in order to execute a subscription. First, the module perform the needed security functions, as the user authentication and access control. Second, the subscription if added to a schedule queue, so the module can execute the subscription while deal with other subscriptions. Next, the module record the subscription data to be used in accounting or other security-related functions. Them, the module query the Privacy Protection Reasoner to create a cloak to the user location information and, after that, the cloaked location is send to the service provider. Finally, the response from the service provider is forwarded to the user.

Since there is the need for an asynchronous communication between the user and the platform, the subscription model should fulfil this requirement. A model that fulfil this requirement is the *Publish-Subscribe*, usually adopted by *Context-aware* architectures, as [21]. This model forces the use of frameworks when developing softwares (a minor penalty in this case, given the novelty and complexity of LBS applications) but create a decoupling between the users and the architecture.

When using the *Publish-Subscribe* model, a middleware need to define a language for the subscriptions. A common approach is the use of languages based on standards as *Extensible Markup Language (XML)* , *Resource Description Framework (RDF)* or *Web Ontology Language (OWL)*, since they provide a good level of interoperability. However,in order to avoid the mobile devices to have to deal with complex languages, a simpler model is adopted where the user only defines his login information, the desired service and his location:

User: login password

Service: service

Location: latitude longitude

Immediately after the module receives a subscription, it's added to a priority queue where each subscription have an associated priority. The scheduling algorithm them execute the subscriptions with higher priorities, switching to the others in order to provide a good throughput and response time, and at the same time avoiding the starvation of other subscriptions.

Besides that functions, the Subscription Manager also have to deal with the accounting of users and services. This function is necessary for the business that would use LBS, as to charge users, control the content and service providers and control the architecture resources. This control (that can be done using logs) also help to increase the privacy and security functions, since the interactions with the architecture are logged, providing a way to verify the entities involved in a privacy or security break.

Since location can be such sensible information, security should be reinforced on a LBS architecture. The Subscription manager module implement the mechanisms for authentication, access control and cryptography of the exchange data taking into considerations the peculiar characteristics of LBS architectures. Since security in LBS is such a complex task and cannot be faced in a brief discussion, this is not focused here, but further information can be found in [29] and [30].

5.3.3 User Profile Repository

The User Profile Repository is the module where the architecture keep all the information about the users. Different from other modules specialized on execute tasks, this module main function is to supply the other modules with the information of the users. This information can vary from login to billing information, but the one focused in this work is the one used to control the privacy of its users, the Privacy Policies.

Privacy Policies are a set of rules where the user define how his personal data can be released. They are modelled using a machine-readable language so, before each information release, the architecture automatically checks if this information can be released. In this architecture, the policies are used to define the rules in which the location of the users is released, and they have the form:

$$\langle service; conditions; action\text{i} \rangle$$

Meaning that, for the service identified by *service* and executed under the *conditions* the action indicated by *action* is performed. The *service* clause is the identifier of the service being invoked, used in the same way as the user invoke the service provider. The *conditions* clauses can be combined in any way to constraint the release of information and can be:

- Time: period of hours.

- Date: period of days.
- Location: area where the user should be located.

Given that there are many types of conditions, the requests can be set-up to cover many situations and provide a fine-grained privacy protection. For each of these situations, the architecture execute one of the following *action*:

- Allow the release with X% of accuracy: the location will be released with a cloak size in which there is X% that the requester can successfully ”guess” the right location. 100% means that the location will be release with no cloak and 0% means that the location will not be released at all.
- Ask user: even though is desirable to the LBS to work without prompting the user, the user can be notified and asked for confirmation if this location can be available to some services.

With privacy rules it's also possible to use Regular Expressions to cover more them one situation. They are checked in the order in which they appear, allowing the creation of simple rules at the same time that hey assure the fine-grained control over the users privacy. This is useful in case of extreme situations, like emergencies, where the users is more willing to allow the release of his information. Follow an example where the location information can be released to the service ”Foo”, between 9:00 to 12:00, with the 60% of probability, and is denied to any other service:

$$\langle Foo; 9:00 - 12:00; 60\% \rangle$$

$$\langle *;*; 0\% \rangle$$

On top of that, for each service that can be more them one rule, so the release of information for the service can be automatically negotiate with the user. If the first rule does not fits the service provider specification, the user can set-up a rule in which he is asked if the released can be done. The Privacy Policy of the last example can be modified in a way that, if 60% of accuracy is not enough to the service provider ”Foo”, the user is asked for confirmation of the release:

$$\langle Foo; 9:00 - 12:00; 60\% \rangle$$

$$\langle Foo; 9:00 - 12:00; ask \rangle$$

$$\langle *;*; 0\% \rangle$$

This type of privacy policy allow the users to create rules in which can deal with the release of location information, taking into consideration who can access his location information, according to the sensibility to the place where he is located and in which situation the user is. The next section show how the Privacy Polices are used to release the location information by the Privacy Protection Reasoner.

What's more, the User Repository keep any information necessary for the user to use the middleware. Security information, as login and password, is a basic need in order to implement security mechanism in the architecture. Billing information is need as well,

to allow business to run on top of the middleware or to pay for the use of some Service Providers and Content Providers. To sum up, all this other information is need to be kept it this module to allow the execution of the services.

5.3.4 Privacy Protection Reasoner

The Privacy Protection Reasoner is the module responsible to define in which way the location of its users is released under a desired level of privacy protection. It received from the Subscription Manager the user privacy preferences and location and, communicating with the Content Provider Proxy, it creates a cloak to the user location. This cloak provides an area where the user is instead of releasing the exact location. The module get from the Content Provider Proxy landscape information from where the user is, computing with the constraints imposed by the landscape and with the user privacy preferences, the size of the cloak. In order to show this computation is done it's necessary to understand the logic behind the reasoning of location privacy.

We considered Data anonymization as a two players game between the potential attacker and the anonymizer, each pursuing his own goal. A prototypical case of data anonymization is location anonymization; here the anonymizer, we will call Bob, protecting the location of a user, we will call Alice, by providing to the potential attacker, from now on called Charlie, some suitably perturbed data, so as to reduce the association of the user to a location. However, as already note in [14] and [20], if Charlie knows the landscape is not neutral, so that some user locations are more likely than others (e.g. due to barriers which constrain the user movements or for the propensity of the user to stay more on specific places, such as her home or her working place), then he could perform some inferences over the data provided by Bob and lower substantially the anonymity level. A second issue to consider [20] - even in a uniform location probability landscape - is that the violation of location privacy can be less or more harmful to the user, depending on the sensitivity of a specific location from the users perspective, and can be less or more profitable to an attacker, depending on the attackers preferences (often the loss taken by the user is proportional to the gain obtained by the attacker).

Hence, in general, an anonymizer needs to take into account possible attacks to the anonymized data, based on both kind of context related information (the location probability landscape and the harm/profitability landscape); the goal of the anonymizer, is minimizing the expected loss to the user, the goal of the attacker will be maximizing his own expected profit. To this aim each player will have the availability of a number of different moves (or pure strategies).

However the outcome of the game for each participant will depend not only on his own choices, but also on the choices of the other player. A player will take a decision based also on his believes about what the other players move will be, and taking into account

what the other player thinks he will do, and so on. This sort of circular interdependence, which for fully rational players brings to an infinite loop, brings the play of cloaking into the territory of Game Theory. If there is a solution it will have to jointly satisfy both players in terms of pay-off. Each strategy pair will correspond to a pair of pay-offs to the two players, a specific pair will be considered a solution to the game if non of the players could gain benefit by leaving that behaviour unilaterally: this solution concept is due to John Nash, and a pair of pure strategies satisfying this condition is called a Nash equilibrium. Depending on the game there can be one or more than one Nash equilibrium solutions, there can even be no solution in pure strategies, in which case a unique Nash equilibrium is granted to exist in mixed strategies: the solution will consist in a suitable randomization among the available pure strategies and will be characterized in a probability distribution over pure strategies. In the cases where the loss suffered upon a successful attack is equal to the pay-off obtained by the attacker, one speaks of zero-sum games, and the players pay-offs can be represented by a single function which one player tries to minimize and the other player to maximize. In those games the solution found by Nash is the same as the solution found earlier by Borel [1], Morgenstern and von Neumann [27], the so called min-max solution: a saddle point of the pay-off function will corresponds to the minimization of one players loss and to the maximization of the other players gain; if there are different solutions they correspond to the same pay-off value, whereas if there are no solutions in pure strategies a single solution is guaranteed to exist in mixed strategies.

The work [20] has framed the problem of data anonymization into the field of game theory by using one of the possible communication scenarios, - we will call ABC, after the initials of the agents and the order the act - where the user Alice act first, asking personalized location related information to Charlie, through the intermediation of the trusted party Bob that send the request along with a cloak to Charlie, who in turn can then deliver an attack. The game between Bob and Charlie was modelled - for simplicity in just one dimension as a two-player zero-sum signalling game (a game with incomplete information for one of the two players, where the uninformed player moves first) and the corresponding equilibrium was characterized. It was found that the game does not have an equilibrium in pure strategies and the solution in mixed strategies was worked out. It came out that when Bob plays the equilibrium strategy the advantage provided to Charlie by the non-neutral landscape gets cancelled: the equilibrium strategy of the attacker consists in attacking uniformly randomly over the cloak and the expected loss to Alice (the expected gain to Charlie) is controlled uniquely by the size of the cloak.

Equally, in the case of this work where Alice, Bob and Charlie represents an user, the middleware and service provider (trying to break the user privacy), this method achieve to protect the user privacy. This is achieved not only in the cases where the service provider is trustful, but also when it’s not and threat the user privacy. Consequently, the middleware provide a method in which not only keep the user’s privacy, but also permit the execution of

the services which need the location information of its users.

5.3.5 Content Provider Proxy

The Content Provider Proxy acts as an abstraction layer between the Content Providers and any entity, as modules or services providers, that need to access geographic content. This content is initially available by the Content Providers, but given that they are provided in several ways, this module provide a single and standardized way to access this data. Therefore, any entity which need to access this data use this module as a simpler way to get the content.

There are many on-line services which provides geographic content in several different ways. There are free services, as Yahoo Maps [11], that provide APIs to various programming languages where a developer can use to create services which need access to geographic content. Others, like GeoNames [5], have on-line databases of geographic content, as street names, available to be used by any service provider. In addition to free services, some services as Google Maps [6] have advanced features available only to the users who pay for the service. The Content Provider proxy deals with the sending of the user billing information and the charging for the content accessed. This myriad of content providers allows various contents to be used by service provides, but increase the complexity as each service has his way to provide the content. The Content Provider Proxy solves this problem making available a single access point to all the content in a standardized way.

The content is accessed in the Content Provider Proxy module through the use of the *Geography Markup Language (GML)* [34]. This is a XML-based language developed by the *Open Geospatial Consortium* [9] and used to the inter exchange of geographical features. Since it's based on XML schemas, it's possible to connect various existing geographical databases, that can have their relational structure define ans XML. For example, a coordinate is represented in GML as:

```
<gml:Point gml:id="p21" srsName="urn:ogc:def:crs:EPSG:6.6:4326">
        <gml:coordinates>34.56, 87.65</gml:coordinates>
</gml:Point>
```

In addition to simple point coordinates, GML can represent high level elements, as roads and rivers. A building can be represented as:

```
<abc:Building gml:id="UnibzMainBuilding">
        <gml:name>Free University of Bozen - Bolzano</gml:name>
        <abc:height>60</abc:height>
        <abc:position>
```

```
                    <gml:Point>
                              <gml:coordinates>46.49,11.35</gml:coordinates>
                    </gml:Point>
          </abc:position>
          <app:extent>
                    <gml:Polygon>
                              <gml:exterior>
                                        <gml:LinearRing>
                                        <gml:coordinates>46.49,11.35</gml:coordinates>
                              </gml:LinearRing>
                    </gml:exterior>
          </gml:Polygon>
     </app:extent>
</abc:Building>
<abc:Building gml:id="UnibzMainBuilding">
     <abc:position xlink:type="Simple" xlink:href="#p21"/>
</abc:Building>
<abc:SurveyMonument gml:id="g234">
     <abc:position>
          <gml:Point gml:id="p21">
                    <gml:coordinates>46.49,11.35</gml:coordinates>
          </gml:Point>
     </abc:position>
</abc:SurveyMonument>
```

The use of GML in the interface of the Content Provider Proxy enable the modeling of geographical data in a general and standardize way, integrating all forms of geographical information.

In addition to the functionality of providing access in a standardized way to other modules, the Content Provider Proxy also have others functions aimed to provide a better service to its users. Since the module is the single point of gathering of content, it also use caching in order to improve the middleware performance. Owing to the fact that the Content Providers are accessed mainly over the Internet, the module keep a copy of the content available to its users. Basically, the cache is updated in the case of request for content that is not already in the cache, or to check if the content actually in the cache is up-todate. Since the module is independent of the other modules, other schemas of caching can be implemented and tested in futher research.

The Content Provider Proxy interact with the Content Providers, The Subscription Manager, Privacy Protection Reasoner and the Service Providers. The Subscription Manager

always request the area which comprises the actual location of an user and the cloak, in order to send together with a service request to a Service Provider. The Privacy Protection Reasoner interact with the module in order to calculate the cloak, requesting information about the landscape where the user is. As explained before, the Content Provider Proxy communicate with the Content Providers performing queries of geographical content. The interaction with the Service Provider is optional, owing to the fact that they can get the content directly with the Content Providers. All the interactions between the modules is explained in details in the section 5.4

5.3.6 Content Providers

The Content Providers are the entities recognisable in the supplying of geographical information related to an user location. Usually, LBS Services Providers don't perform they services based purely on the user location, but they also need other data related to where the user is and what he is trying to accomplish. What's more, geographical information is a complex data, with high complexity and that needs special infrastructure, as GIS databases and access ways. For these reason they need special entities, the Content Providers, to deal with this complexity. Not really from the architecture, just described here to understand the execution of the middleware.

The Content Providers can make information available in various ways, from dynamic content accessed through an API to a static data available in a website. Services as Google Maps cite and Yahoo! Maps cite have public APIs with varying functions and possibilities of use. The Geonames project cite keep a database of geographical information available in their websites, free for use by developers. Some Content Providers charge for their services, depending on the content required, or need a previous registration before the use. These singularities make each Content Provider a different entity to be used in different way.

The Content Providers are accessed by the Content Provider Proxy for the gathering of content. The Content Provider Proxy query the Content Providers, getting the necessary information for the execution of a given service. They are not accessed only when needed but also to populate the the Content Provider Proxy caching system. But this module is not accessed only by the Content Provider Proxy.

The Services Providers can access the Content Provider directly, instead from the Content Provider Proxy, for some reasons. One of the reasons is the fact that an attacker posing as a Service Provider may not want the middleware to know what kind of information related to the user location the attacker needs. This decreases the amount of information related to an attack to the user is available to the platform. Another reason is that a Service Provider can have a private Content Provider, not available to other entities. For these reasons, even though is not expected, the Content Providers can allow access to these entities.

5.3.7 Service Providers

The Services Providers are the entities responsible for supplying LBS services to the users of the platform. They act in the end of the LBS supply chain, providing services requested by the users, intermediated by the middleware. Usually, these services providers just use this location to provide the services to the users, but they can also use it to threat the users privacy. The Service Providers are not situated inside in the architecture, but since it provide services to the middleware users, it is described here in order to help to understand his interaction with the middleware.

The Services Providers interact with the middleware in two ways: through the supplying of service to the users of the platform and through the use of the content provided by the Content Provider Proxy. In the first way, the Service Providers receive the service request, execute a specific LBS service and return the response to the platform, that forward this response to the user which requested the service. The user doesn't request the service directly, but request it through the platform, that receive the user location, cloak it and them forward it to the Service Provider. The other way to interact with the middleware is through the gathering of content from the Content Provider Proxy. Since the Service Providers need geographical content, they can get this data from the Content Provider Proxy module or contact directly some Content provider.

By definition, the Service Providers need to gather geographical content in order to perform a service to an user. The main type of information needed is the user location, but usually a Service Provider also need information related to the user location, as nearby *Points of Interest (POI)*. Examples of Service Providers are Tomtom, which use traffic and weather information related to the user location, and Loopt, that use the user location to get nearby friends and interesting paces (in details in the section 2.4). On the other hand, there is no guarantee that a Service Provider use the user location only to perform the requested service. They can perform a service to an user but also use the location information in unexpected ways.

Attackers can pose as Services Providers and use the location available of an user to threat his privacy. Given that all Service Providers are entities external to the architecture and don't have a trust relationship with the middleware, they cannot be assumed that they use the users location only to perform a service. Location information can be used to a wide range other acts, from annoying unwanted advertisements spread by spammers to physical harm done by people with different political views or lifestyles. To avoid this problem, the middleware doesn't send the exact same location that the user send to the middleware, but a cloaked one. The privacy protection mechanism assume that they can threat the user privacy, but since they provide some desired service, the information is send in a way to achieve both aims: execute the service and protect the user privacy. This approach guarantee that the service is executed without creating threats to the privacy of the user.

Given that the location send to the Services Providers is not a exact point in the space, but an area, sometimes this can lead to problems in the execution of the service. Some Service Providers may need a more precise location or they are unable to operate. In this case, a Service Provider can resend the request for location and, if the user have a secondary rule in his Privacy Policy, he can be asked to allow a location with a smaller cloak. This can be used in cases where the user trust more in a specific Service Provider or have a urgent need to execute this service in that moment. This strategy allow the execution of the service without creating a threat to the user privacy.

5.4 MY MIDDLEWARE OPERATION

The execution of a LBS by an user follows a specific order of interactions between the modules in the architecture. For the user, Content Providers and Service Providers, the interactions between the modules are transparent, i.e. they are only aware of their interactions with the architecture. For example, when an user request a service to a LBS, he don't know which Content Providers provide the geographical content needed for the execution of the service. The iteration schema, seen in the figure 6, is regard an user invoking a LBS and follow the steps:

1. The User request a Service Provider sending a subscription to the middleware (the Subscription Manager receives the subscription), informing his login information, the desired Service Provider and his location.
2. The Subscription Manager request the user profile to the User ProfileRepository, sending his login information.
3. After the User Profile Repository successfully authenticate the user, hereturn the user profile to the Subscription Manager.
4. With the user privacy preferences get from the user profile, the Subscription Manager request the cloak size to the Privacy Protection Reasoner, sending the user location and privacy policy.
5. The Privacy Protection Reasoner request information about the landscapesurrounding the user location to the Content Provider Proxy, as streets and buildings.
6. The Content Provider Proxy returns the requested information to thePrivacy Protection Reasoner.
7. With the privacy preferences and the landscape information, the PrivacyProtection reasoner compute the size of the cloak and send it to the Subscription Manager
8. Them, the Subscription Manager request a map with the characteristicsof the cloak to the Content Provider Proxy, in order to be send to the Service Provider.
9. The Content Provider Proxy return a map with the given characteristicsto the Subscription Manager
10. The Subscription Manager send this map, instead of the user exact location, to protect the user privacy, invoking the desired Service Provider.
11. The Service Provider execute his service and send a service response tothe

Subscription Manager.

12. The Subscription Manager forward the service response to the user whichhave requested the service.

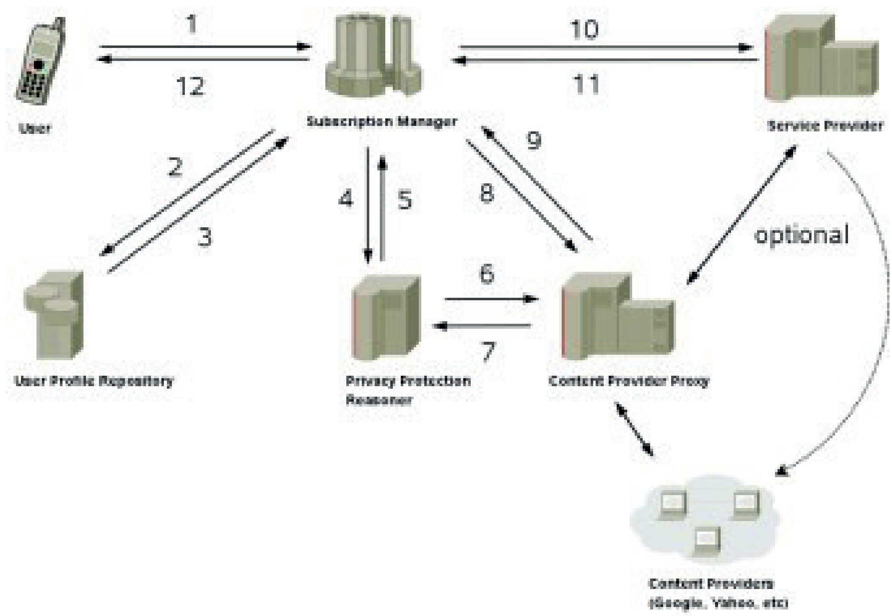


Figure 6: Middleware operation

The execution don't take into consideration the interactions between the Service Providers, the Content Providers and the Content Providers Proxy. The Service Providers can interact with these other entities in order to gather content to provide the desired service. This happens specially when a Service Provider is an attacker that perform inferences to threat the user privacy. In this case, the Service Provider may gather content directly from the Content Providers, avoiding in this way to make the middleware aware of the information that the attacker is using to infer the user location. All these interactions can happen in parallel to any of the interactions between the modules in the architecture.

5.5 KEY STUDY

Key study using real world data about cars insurance's (check with the presentation) one example where the guy is bad, other where he is good usar imagens de bolzano do google earth

CONCLUSION AND FUTURE WORK

CONCLUSION

While some computer applications help us to perform tasks in an easier and better way, some amaze us with the possibility to do unimaginable things before. The development of mobile devices and wireless networks allow the appearance of one of this kind of application, the Location-Based Services. It not only allows the devices to wirelessly communicate with others, but also to be aware of the surrounding objects and provide functionalities that use this information. Location-Based Services certainly change the way that daily tasks are performed.

This work was the initial proposal for the middleware. This proposal include the model of interaction between the architecture, users, service providers and content providers, in addition to the modules which constitute the architecture itself. Every module have its special functionalities that aim to satisfy all the requirements imposed by LBS applications and studied in the first chapters. These requirements fulfilment and the architecture design were specially approached in order to support the protection of the users privacy from inference attacks. This characteristic secure their interactions with the LBS and distinguish the middleware from regular solutions.

FUTURE WORKS

Due to the fact that LBS are a complex area, it's impossible in a single work to cover all aspects involved in this field. Some points were not deliberately deeply studied here, constituting some subjects for futher research. The subscription language user by the Subscription Manager, even though is suitable to the current needs, can be extended to allow different types of interactions with the middleware. The Privacy Protection Reasoner should be implemented in order to provided a concrete software where the privacy protection can be tested. Also, the underlying security mechanism needed by LBS systems constitute a wide research area. These facts open the opportunity for futher research in some points.

In addition, some LBS services and applications can be created in order to test the various aspects of the middleware. After the development of the Privacy Protection Reasoner, the middleware can be tested with real-world data, checking the level of privacy-protection provided by this approach. Also, tenting the Content Provider Proxy will allow the definition of better-suited caching algorithms. These applications will also serve to check the speed in which the applications can run on the middleware, identifying bottle-necks and possible changes in the technology.

REFERENCES

Traite du Calcul des Probabilites et ses Applications. Gautier-Villars, 1938.

Java 2 platform, micro edition (j2me) web services. Technical report, Sun Microsystem, 2004.

Cospas-sarsat system overview, June 2005. http://www.equipped.org/cospas-sarsat overview.htm.

Location-based Services: Fundamentals and Operation, chapter 6, 7, 8 and 9. John Wiley & Sons, 2005.

Geonames, July 2008. http://www.geonames.org/.

Google maps api, July 2008. http://code.google.com/apis/maps/.

Half the world will use a cell phone by 2009, August 2008. http://www.mobiledia.com/news/43104.html.

Loopt, July 2008. http://www.loopt.com/.

Open geospatial consortium inc., July 2008. http://www.opengeospatial.org/.

Tomtom, July 2008. http://www.tomtom.com/.

Yahoo! maps api, July 2008. http://developer.yahoo.com/maps/.

Westin A. *Privacy and Freedom*. Atheneum, New York, 1970.

Claudio Ardagna, Marco Cremonini, and Gabriele Gianini. Landscapeaware location-privacy protection in location based services. *Journal of Systems Architecture*, to appear 2008.

Marco Cremonini Claudio Ardagna and Gabriele Gianini. Reference number of this ogc project document: Ogc 07-113r1. Technical report, Open geospatial consortium inc, 2007.

A.R. Beresford and F. Stajano. Location privacy in pervasive computing. *Pervasive Computing, IEEE*, 2(1):46–55, Jan-Mar 2003.

A. O. Cruz. Um estudo sobre controle de privacidade em plataformas de servi¸cos sens´ıveis ao contexto. *Universidade Federal do Esp´ırito Santo*, 2006.

Gianpaolo Cugola and H.-Arno Jacobsen. Using publish/subscribe middleware for mobile systems. *SIGMOBILE Mob. Comput. Commun. Rev.*, 6(4):25–33, 2002.

Anind K. Dey. Understanding and using context. *Personal Ubiquitous Comput.*, 5(1):4–7, 2001.

European Commission. *Directive 95/46/ec of the european parliament and of the council of 24 october 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data*, November 1995.

Gabriele Gianini and Ernesto Damiani. A game-theoretical approach to data-privacy protection from context-based inference attacks: A locationprivacy protection case study. In *Proc. of the workshop on Secure Data Management SDM 2008, LNCS 5159*, pages 133–150, Berlin Heidelberg, 2008. Springer-Verlag.

Bernardo Gonc¸alves, Jos´e G. Pereira Filho, and Giancarlo Guizzardi. A service architecture for sensor data provisioning for context-aware mobile applications. In *SAC '08: Proceedings of the 2008 ACM symposium on Applied computing*, pages 1946–1952, New York, NY, USA, 2008. ACM.

M. Gruteser and Xuan Liu. Protecting privacy in continuous locationtracking applications. *Security & Privacy, IEEE*, 2(2):28–34, Mar-Apr 2004.

Marco Gruteser and Dirk Grunwald. Anonymous usage of location-based services through spatial and temporal cloaking. In *MobiSys '03: Proceedings of the 1st international conference on Mobile systems, applications and services*, pages 31–42, New York, NY, USA, 2003. ACM.

H-.A Jacobsen, editor. *Middleware Services for Selective and LocationBased Information Dissemination*, Advanced Topic Workshop on Middleware for Mobile Computing, November 2001. 12-16.

Jae-Chul Kim. *On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops*, chapter Middleware Platform for Ubiquitous Location Based Service, pages 965 – 973. 2006.

Axel Kupper. *Location-based Services: Fundamentals and Operation*. John Wiley & Sons, 2005.

Oskar Morgenstern and John Von Neumann. *Theory of Games and Economic Behavior*. Princeton University Press, May 1944.

G. Myles, A. Friday, and N. Davies. Preserving privacy in environments with location-based applications. *Pervasive Computing, IEEE*, 2(1):56–64, Jan-Mar 2003.

Namje Park, Howon Kim, Kyoil Chung, and Sungwon Sohn. A secure and privacy enhanced lbs security elements based on klp. *Geoscience and Remote Sensing Symposium, 2005. IGARSS '05. Proceedings. 2005 IEEE International*, 2:1221–1224, July 2005.

Namje Park, Kiyoung Moon, Howon Kim, Kyoil Chung, and Sungwon Sohn. An efficient software-based security acceleration methods for open lbs services. *Geoscience and Remote Sensing Symposium, 2005. IGARSS '05. Proceedings. 2005 IEEE International*, 1:4 pp.–, July 2005.

Cuellar J. R. *Geographic Location in the Internet*, chapter Location Information Privacy, pages 179–208. Kluwer Academic Publishers, 2002.

Jochen H. Schiller and Agn`es Voisard, editors. *Location-Based Services*. Morgan Kaufmann, 2004.

A. Schmidt and K. van Laerhoven. How to build smart appliances? *Personal Communications, IEEE [see also IEEE Wireless Communications]*, 8(4):66–71, Aug 2001.

Ron Lake Simon Cox, Paul Daisey. Opengis geography markup language (gml) - implementation specification. Technical report, Open Geospatial Consortium, Inc., 2004.

Fred Stutzman. The new presence, November 2007. http://dev.aol.com/article/2007/the new presence.

K. Virrantaus, J. Markkula, A. Garmash, V. Terziyan, J. Veijalainen, A. Katanosov, and H. Tirri. Developing gis-supported location-based services. *Web Information Systems Engineering, 2001. Proceedings of the Second International Conference on*, 2:66–75 vol.2, Dec 2001.

Mark Weiser. The computer for the twenty-first century. *Scientific American*, 265(3):94–104, 1991.

Z Fan Z Wang X Zou, X Tang. Research of lbs based on java and an application solution. In *The Industrial Demonstration of LBS in Digital Haerbin*, Key Laboratory of Geo-informatics of State Bureau of Surveying and Mapping - Chinese Academy of Surveying and Mapping. Commission II, WGII/1,2,7 and Commission VII, WG VII/6.

ABOUTH THE AUTHOR

ADILSON OLIVEIRA CRUZ, a Linhares, Brazil native born in 1980, has always had a strong affinity for technology. His educational journey includes a Technical Degree in Data Processing from Centro Federal de Educação Tecnológica do Espírito Santo (CEFET/ES), a Bachelor's Degree in Computer Science from the Universidade Federal do Espírito Santo (UFES), and a Master's Degree in Informatics from Libera Università di Bolzano/Bozen in Italy.

Adilson has accumulated significant experience in the fields of computer science and technical and higher education. He is known for his contributions to both academia and the professional world, demonstrating a commitment to advancing technology and educating the next generation.

In addition to his professional pursuits, Adilson shares his knowledge through his written work, solidifying his reputation in the local tech industry. Moreover, Adilson actively contributes to the local education landscape, further cementing his dedication to nurturing talent in his community.

Adilson Oliveira Cruz's journey serves as a source of inspiration and knowledge, making him a author who has made a impact in the realms of computer science and education.

A LBS MIDDLEWARE WITH PRIVACY PROTECTION FROM INFERENCE ATTACKS

www.atenaeditora.com.br
contato@atenaeditora.com.br
@atenaeditora
www.facebook.com/atenaeditora.com.br

Atena Editora
Ano 2023

A LBS MIDDLEWARE WITH

PRIVACY PROTECTION

FROM INFERENCE ATTACKS

www.atenaeditora.com.br
contato@atenaeditora.com.br
@atenaeditora
www.facebook.com/atenaeditora.com.br

Atena Editora
Ano 2023